

## **REMARKS**

Applicant is in receipt of the Office Action mailed October 2, 2007. Claims 1 and 6-23 are pending in the case. Reconsideration of the present case is earnestly requested in light of the following remarks.

### **Section 103 Rejections**

Claims 1 and 16-23 were rejected under 35 U.S.C. 103(a) as being unpatentable over Deutscher et al (US Patent Pub. 2004/0001106, "Deutscher") in view of Hampapuram et al. (US Patent Pub. 2004/0221262, "Hampapuram"). Applicant respectfully traverses the rejection.

Claim 1 recites:

1. A memory medium which stores program instructions implementing a graphical user interface (GUI) for debugging a program, wherein, during execution of the program, the program instructions are executable by a processor to perform:

displaying source code for the program on a display during execution of the program;

receiving first user input hovering a mouse cursor over an expression in the source code;

in response to said hovering the mouse cursor over the expression, displaying a GUI element proximate to the expression, wherein the GUI element includes a value of the expression;

receiving second user input to the GUI element modifying the displayed value, thereby specifying a new value for the expression; and

setting the expression in the program to the new value in response to the second user input, wherein the program continues execution in accordance with the new value of the expression.

Nowhere does Deutscher teach or suggest **receiving first user input hovering a mouse cursor over an expression in the source code or setting the expression in the program to the new value in response to the second user input, wherein the program continues execution in accordance with the new value of the expression** as recited in claim 1.

As summarized in paragraph 27, Deutscher is directed to a presentation production program, also referred to as a production tool, that in response to user input generates a presentation file that is executable by a viewer to display a presentation. More specifically, the user selects a template implementing a desired presentation schema, and selects or creates scripts specifying presentation scheduling for multi-media content, where the scripts are executed as part of the template.

Scripts are called from a scripts grid, which, per paragraph 137, is a scheduling table where the user enters events or script commands that will be triggered during playback of the presentation. Per cited paragraph 143, to edit the scripts grid, the user *double clicks on the event line* of the script grid to be edited, invoking a GUI for the parameters of the script whereby the user may change the values of the parameters, as illustrated in Figure 13.

Applicant respectfully submits that the cited script grid being edited in Deutscher is not a program, but, as described in paragraph 137, is “essentially a scheduling table where the user enters events (sometimes referred to as script commands) that will be triggered during the playback of the presentation”. Nowhere does Deutscher describe this table as an executable or executing program. Applicant respectfully submits that the script grid is more properly described as a data file, noting that, per Deutscher, paragraphs 137-144, the script grid lists script names, along with their timing data and parameter values, but does not include the scripts themselves, which are executable. Nowhere does Deutscher describe the script grid as an executable program. Thus, while in Deutscher, the user may invoke a GUI to add, modify, or configure execution of a script from the script grid, displaying the script grid does not in fact display source code, e.g., of the scripts. More specifically, since the script grid is not an executable program, and the event lines are not executable expressions, Deutscher’s user selection of an event line (which comprises data, not program code) does not teach or suggest displaying

*source code for a program on a display during execution of the program, nor receiving first user input hovering a mouse cursor over an expression in the source code, nor setting the expression in the program to the new value.*

Per paragraph 137, the program that is paused for Deutscher's script grid editing is "a video or audio program that is designated as the master track". Nowhere does Deutscher describe displaying the source code or editing expressions in the source code of this program, as recited in claim 1.

The Office Action admits that Deutscher fails to mention "hovering a mouse" over an expression to invoke a GUI for editing the expression in source code, but then asserts that since double-clicking and hovering are both well-known mouse events, hovering the mouse cursor over an expression to invoke the GUI to edit the expression is an obvious variant of Deutscher's disclosed double-clicking on an event line to configure a script. Applicant respectfully disagrees.

In paragraphs such as 140, 143, and 201 (and elsewhere), Deutscher describes editing a script grid, e.g., via a user *double-clicking* on an event line in the script grid to invoke a GUI for specifying and configuring execution of a script indicated in the event line. Applicant respectfully submits that a novel and useful aspect of Applicant's invention as represented in claim 1 is to provide a *tooltip* with the claimed features, specifically, a tooltip that allows an expression in the source code of *an executing program* to be edited through the tooltip. This technical feature is not provided by the prior art technique of *double-clicking* on an event line (specifying a script) to invoke a GUI for configuring execution of the script, as taught by Deutscher. As is well-known, a tooltip is particularly characterized by *not* requiring a user to double-click on an expression with a mouse, and so Deutscher actually teaches away from the invention as represented in claim 1.

Moreover, Applicant respectfully submits that double-clicking and (prior art) mouse hovering being well-known in no way means that mouse cursor hovering is an obvious variant of double-clicking, and additionally, respectfully submits that since a feature of claim 1 is a *tooltip* with the claimed editing functionality, it is improper for the Examiner to ignore this important and novel feature of the claim. Additionally, Applicant notes that Deutscher nowhere discusses or even hints at hovering a mouse

cursor, and respectfully submits that a reading of Deutscher would not compel one of ordinary skill in the art to contemplate invoking a GUI for configuring a script of an event line by hovering a mouse cursor over the event line.

Thus, for at least these reasons, Applicant submits that the cited art fails to teach or suggest this feature of claim 1.

Nor does Deutscher disclose **in response to said hovering the mouse cursor over the expression, displaying a GUI element proximate to the expression, wherein the GUI element includes a value of the expression, and receiving second user input to the GUI element modifying the displayed value, thereby specifying a new value for the expression**, as recited in claim 1.

As discussed above, Deutscher fails to disclose hovering a mouse cursor over an expression at all, does not teach hovering or double clicking on source code of a program, and so does not and cannot teach displaying a GUI element (that includes a value of the expression) proximate to the expression *in response to hovering a mouse cursor over the expression*. Applicant respectfully notes that Hampapuram nowhere describes the user editing a value of an expression via a tooltip, and so is not germane to claim 1.

Thus, for at least the reasons provided above, the cited art fails to teach or suggest this feature of claim 1.

Thus, for at least the reasons provided above, Applicant submits that the cited art of Deutscher and Hampapuram, taken singly or in combination, fails to teach or suggest all the features and limitations of claim 1, and so claim 1, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over the cited art, and are thus allowable.

Independent claims 18-21 each includes similar limitations as claim 1, and so the above arguments apply with equal force to these claims. Thus, for at least the reasons discussed above, Applicant submits that claims 18-21, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over the cited art, and are thus allowable.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown

to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

Removal of the section 103 rejection of claims 1 and 16-23 is earnestly requested.

## **CONCLUSION**

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5150-82801/JCH.

Also filed herewith are the following items:

- ☐ Request for Continued Examination
- ☐ Terminal Disclaimer
- ☐ Power of Attorney By Assignee and Revocation of Previous Powers
- ☐ Notice of Change of Address
- ☐ Other:

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198  
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8800  
Date: 2007-12-18 JCH/MSW